NORTH-HOLLAND

Author Index

A
Abdel Mageed, M. Z., 31:239
Ahrens, J. D., 30:71
Arafeh, B. R., 31:7
Atkinson, D. C., 30:99

B
Balci, O., 31:215
Bellman, K. L., 29:199
Berry, D. M., 28:179
Bieman, J. M., 28:19, 30:271
Breslawski, S., 31:135
Brown
  A. W., 28:59
  L., 29:149

C
Caruso, J., 30:45
Chang, R.-C., 29:177
Chatman, V. V. III, 31:71
Chow, C.-H. E., 28:253
Chung, S.-L., 31:267
Cimitile, A., 28:117
Coleman, D., 29:3
Conradi, R., 30:201
Costello, R. J., 29:39

D
Davis, A. M., 30:151
Derrick, E. J., 31:215
Dhama, H., 29:65
Dutertre, B., 28:77
Duvall, L. M., 31:109

E
Eassa, F. E., 31:239
Evanco, W. M., 29:75

F
Fang, M., 28:239
Fletcher, P., 30:161
Fox, C. J., 30:295
Frakes, W. B., 30:295
Fritzson, P., 31:55

G
Girardi, M. R., 30:249
Glass, R. L., 28:1, 28:3, 28:93, 28:187, 29:95, 30:160, 31:1, 31:3, 31:95
Gomaa, H., 28:189
Gonzalez, R. R., 29:17
Griss, M. L., 30:213
Griswold, W. G., 30:99

H
Ha, R., 30:85
Harris, D. P., 29:191
Harrison
  R. D., 30:45
  W., 29:1
Hayne, S., 28:97
Heim, M., 28:89
Heinz, E. A., 28:9
Henninger, S., 30:231
Henry, J., 28:169
Hetzel, W. C., 31:171
Ho-Stuart, C. J., 28:239
Hops, J. M., 31:157
Hwang, M.-S., 31:257

I
Iannello, G., 29:107
Ibrahim, B., 30:249
Isoda, S., 28:49, 28:69, 30:171

K
Kamkar, M., 31:55, 31:197
Kandelin, N. A., 29:261
Karlsson, E.-A., 30:201
Karunanithi, S., 30:271
Khoshgoftaar, T. M., 29:85
Klein, G., 28:39
Kobrosly, W., 28:129
Kung, D. C., 31:33
Kuroki, H., 28:49

L
Lai, R., 28:143
Lam, K. Y., 31:267
Landauer, C., 29:199
Lanning, D. L., 29:85
Leary, J. R., 30:27
Lederer, A. L., 31:125
Lennartsson, B., 29:92
Li, J.-P., 29:135
Littlejohn, K., 30:117
Liu
  D.-B., 29:39
  J. W. S., 30:85
  S., 28:77, 29:167
Lock, E., 30:71
Louis, R. C., 28:157
Lowther, B., 29:3
Lu, P.-K., 29:177
Lukowicz, P., 28:9

M
Mackey, S. R., 30:137
Masters, M. W., 30:45
Mathur, A. P., 31:185
Mazzeo, A., 29:107
Mazzocca, N., 29:107
Mehrotra, M., 29:235
Meredith, L. M., 30:137

J. SYSTEMS SOFTWARE 1995; 31:279–280
© 1995 by Elsevier Science Inc.
655 Avenue of the Americas, New York, NY 10010

0164-1212/95/$9.50
SSDI 0164-1212(95)00169-7

Morris, E. J., 28:59
Mostert, D. N. J., 31:45
Murrell, S., 29:251
Mutka, M. W., 29:135

O

Oka, A., 28:49, 28:69
O'Leary, D. E., 29:261
Oman, P., 29:3
Osterweil, L. J., 31:239

P

Pfleeger, S. L., 31:143
Plant, R., 29:197, 29:251
Poulin, J. S., 30:187
Prasad, J., 31:125
Prechelt, L., 28:9
Preece, A. D., 29:219
Prywes, N., 30:71

R

Rader, J. A., 28:59
Ram, S., 29:97
Rossman, A., 28:169

S

Sage, A. P., 30:3
Samadzadeh, M., 30:167
Samuel, A. L., 30:45
Sanden, B. I., 28:225
Semmel, R. D., 30:1, 30:127
Shahmehri, N., 31:55
Sherer, S. A., 29:97
Sherif, J. S., 31:157
Sindre, G., 30:201
Snyder, J., 28:169
Sobol, M. G., 28:39
Stavridou, V., 28:77
Stoen, R., 29:121
Subramanian, G. H., 31:135

T

Takahashi, K., 28:69
Tang, S. M., 28:157
Tesch, D. B., 28:39
Thambu, P., 28:267, 29:121
Tichy, W. F., 28:9
Triantafyllos, G., 28:129
Turley, R. T., 28:19

U

Ulusoy, O., 31:97
Urban, S. D., 28:203

V

Vassiliadis, S., 28:129
Visaggio, G., 28:117
von Solms, S. H., 31:45

W

Wang
A. M., 28:203
H.-H., 29:177
Welch, L. R., 30:45
Wentzel, K. D., 30:213
Wild, C., 29:235
Wilkening, D. E., 30:117
Wilson, M., 30:1, 30:45
Winkler, R. P., 30:127
Wong
J., 28:267, 29:121
W. E., 31:185
Wu, J., 29:149

Y

Yamamoto, S., 28:49, 28:69
Yang, W.-P., 31:257

Z

Zand, M., 30:167
Zedan, H. S. M., 28:239

NORTH-HOLLAND

Subject Index

A

Academic/industrial collaboration in software measurement, 29:1
Ada programs, dynamic analyzer for, 31:239
AIDA, Ada programs and, 31:239
Algorithms
  efficient token-based mutual exclusion, 28:267
  encryption data base security, 31:257
  fault-tolerant distributed shared memory systems, 29:149
  mutual exclusion in distributed system, 29:121
  priority-driven, 30:85
  real-time constraints validation, 30:85
  scheduling, 30:85
Allocation, resource, for multiparty connections, 28:253
Application layer protocol, automatic implementation of, 28:143
Architectural approaches, for software systems, 30:27
Authors, leading, in software engineering, 31:3
Automation
  fused data bases and automated query formulation, 30:127
  implementation of application layer protocol, 28:143
  usability criteria for automated debugging systems, 31:55
  visual simulation support environment, DOMINO conceptual framework and, 31:215

B

Behaviors, timed-probabilistic behaviors model, 28:239
BSD UNIX, external pager facility for memory management capability, 29:177

C

Call dominance tree, software salvaging and, 28:117
CASE integration, operational use, 28:59
CASE tools
  information system professionals' attitudes on, 28:39
  structured methodology and, 28:49
CHANGE-POINTs, software productivity measurement with, 31:71
Citation frequency, research quality and, 30:160
Client domain, requirements engineering and, 28:179
Cohesion and coupling, quantitative model of software, 29:65
Competency, software engineer, 28:19
Complex systems, reengineering for, 30:1, 30:45
Complexity models, and metric for software maintenance, 31:157
Computer safety, requirements specification and, 31:45
Computer security, safety, and resilience requirements, requirements specification and, 31:45
Computing research, structure-based critique, 28:3
Conceptual modeling, object-oriented, 31:33
Concurrency, reengineering complex systems and, 30:45
Concurrent software, specification of distributed and, 31:7
Constraint/rule language design, for object-oriented data model, 28:203
Constraints acquisition technique, in requirements specification, 31:45
Cost estimates, inaccurate, 31:125
Coupling, quantitative model of software cohesion and, 29:65

D

Data analysis software, for system security, 31:267
Data bases
  constraint/rule language design for object-oriented data model, 28:203
  distributed real-time data base systems, transaction- processing architectures for, 31:97
  encryption scheme for enhancing security, 31:257
  fused, 30:127
  group design, 28:97
Debugging systems, usability criteria for automated, 31:55
Decision making
  metric-based, 31:95
  quantitative, 31:95
Distributed memory computer, parallel node architecture in, 29:107
Distributed real-time data base systems, transaction- processing architectures for, 31:97
Distributed shared memory systems, snooping fault-tolerant, 29:149
Distributed software, specification of concurrent and, 31:7
Distributed system
  algorithm, efficient token-based mutual exclusion, 28:267
  mutual exclusion algorithms in, 29:121
Domain-dependent/independent approaches, for object-oriented system verification, 29:261
Domain models, reusable software requirements and architectures, 28:189
Domain selection/analysis, in software reuse, 30:171
Domain-specific kits, software reuse and hybrid, 30:213
Domain-specific software, reusable software libraries and, 30:187
Dominance relation, call-directed graph nodes and, 28:117
DOMINO conceptual framework, visual simulation support environment and, 31:215
DRMS, real-time tasks allocation, 29:135
Dynamic analyzers, for Ada programs, 31:239
Dynamic testing tools, for Ada programs, 31:239

E

Education, software engineering, 29:92
Encryption scheme, data base security enhancement, 31:257
Enhancement, software maintainability and, 29:75
Entity-life modeling, designing control systems with, 28:225
Environmental adaptations, software maintainability and, 29:75
Error prediction, static error prediction models, 28:129
Estimate alterations, software effort, 31:135
Experimental validation, 28:9
Experiments, role in computer science, 30:161
External pager facility, memory management capability with, 29:177

F

Fault correction, software maintainability and, 29:75
Fault detection, software maintainability and, 29:75, 29:85, 29:97

J. SYSTEMS SOFTWARE 1995; 31:281–284
© 1995 by Elsevier Science Inc.
655 Avenue of the Americas, New York, NY 10010

0164-1212/95/$9.50
SSDI 0164-1212(95)00170-0

Fault-tolerant distributed shared memory systems, algorithms for, 29:149
Formal methods
formal requirements specification method, 29:167
for safety-critical systems, 28:77
software engineering, 29:95
Formal requirements specification method, internal consistency with, 29:167
Formal semantics, for rule-based systems, 29:251

G
Graph grammer model, for specification of distributed and concurrent software, 31:7
Group view modeling, group data base design and, 28:97

H
Hashing, letter-oriented hashing schemes, 28:157
High-risk module detection, software maintainability and, 29:85
Hybrid domain-specific kits, for software reuse, 30:213

I
Industrial collaboration, software measurement and academic and, 29:1
Industrial software systems, software maintainability models in, 29:3
Information access tools, for software reuse, 30:231
Information architecture, architectural approaches for software systems, 30:27
Information system development, professionals' attitudes, 28:39
Institutions, leading, in software engineering, 31:3
Intrusion detection, 31:267
ISO ACSE protocol, automatic implementation of, 28:143

J
Journal of Systems and Software, publication policy changes, 28:1

K
Key collision problems, letter-oriented hashing schemes and, 28:157
Knowledge-based systems
multiviewpoint clustering analysis in, 29:235
quality assessment framework for, 29:219
software quality in, 29:197
verification/validation methods for, 29:199

L
Language design, constraint/rule, for object-oriented data model, 28:203
Layering, reengineering complex systems and, 30:45
Leading scholars/institutions, in software engineering, 31:3
Legacy software, reengineering, 30:71
Letter-oriented hashing schemes, 28:157
Libraries, reusable software, 30:187, 30:201
Logical connection, resource allocation for multiparty connections and, 28:253

M
Maturity, in metrics plan, 31:143
Measurement, unified metric of software complexity, 29:17
Memory management capability, external pager facility for, 29:177
Mentoring/patterning approach, research, 28:3
Metric-based decision making, 31:95
Mission-critical software architecture, reengineering and, 30:45
Multiparty connections, resource allocation for, 28:253
Multiviewpoint clustering analysis, in knowledge-based systems, 29:235
Mutation testing, cost reduction, 31:185
Mutual exclusion algorithms
distributed system, 29:121
efficient token-based, 28:267
Mythseeker project, 28:89

N
Natural language, for software retrieval, 249
Neural network approach, software maintainability and, 29:85, 29:97

O
Object-based programming languages, language-supported reuse in, 30:271
Object-oriented conceptual modeling, visual formalism for, 31:33
Object-oriented data model, constraint/rule language design for, 28:203
Object-oriented design, transition from object-oriented requirements, 30:151
Object-oriented programming languages, language-supported reuse in, 30:271
Object-oriented requirements, transition to object-oriented design, 30:151
Object-oriented system verification, domain- dependent/independent approaches for, 29:261

P
Parallel node architecture, distributed memory computer with, 29:107
Personnel
information system professionals' attitudes, 28:39
software engineers' competencies, 28:19
Probability, timed-probabilistic behaviors model, 28:239
Productivity measurement
software, 31:71
unified metric of software complexity, 29:17
Program reliability, CASE method and, 28:49
Program slicing, classification of, 31:197
Program understanding and transformation tool, for software maintenance, 30:99
PROMPT, for automatic implementation of application layer protocol, 28:143
Protocol, application layer, automatic implementation of, 28:143
Publication policy changes, *Journal of Systems and Software*, 28:1

Q
Quality assessment framework, for knowledge-based systems, 29:219
Quality measurement, unified metric of software complexity, 29:17
Quantitative decision making, 31:95
Query formulation, fused data bases and automated, 30:127
QUICK, for data base reverse engineering, 30:127

R
Rate-monotonic scheduling technology, tool for applying, 29:135
Real-time constraints, validation methods for, 30:85
Real-time tasks, allocation, 29:135
REBOOT, software reuse, 30:201
Requirements engineering
client domain and, 28:179
metrics for, 29:39
Requirements specification
computer security, safety, and resilience requirements in, 31:45
internal consistency with, 29:167
Research
computer science experiments, 30:161
experimental validation, 28:9
qualitative, 30:160
structure-based critique, 28:3
structured approaches, 28:93
Resilience requirements, requirements specification and, 31:45
Resource allocation, for multiparty connections, 28:253
Retrieval tools, for software reuse, 30:231, 30:249
Reverse engineering, data fusion and, 30:127
ROSA, software reuse system, 30:249
Rule-based systems, formal semantics for, 29:251

S

Safety-critical systems, formal methods practice, 28:77
Scheduling, real-time tasks allocation, 29:135
Scholars, leading, in software engineering, 31:3
Security
  encryption scheme for data base, 31:257
  multivariate data analysis software for system, 31:267
Semantics
  formal, for rule-based systems, 29:251
  software maintenance and, 30:99
Simulation model, DOMINO conceptual framework and visual, 31:215
Slicing, program, classification of, 31:197
Snooper, fault-tolerant distributed shared memory systems, 29:149
Software approaches, types, 28:187
Software architectures, reusable software requirements and, 28:189
Software complexity, unified metric of, 29:17
Software costs, cohesion and coupling quantitative model, 29:65
Software design
  designing control systems with entity-life modeling, 28:225
  experimental validation, 28:9
  object-oriented requirements to object-oriented design, 30:151
Software development
  CASE tools with structured methodology, 28:49
  effort estimate alterations, 31:135
  formal methods in safety-critical systems, 28:77
  formal requirements specification method, 29:167
  inaccurate cost estimates, 31:125
  maintainability in industrial software systems, 29:3
  management, 31:109
  metrics for requirements engineering, 29:39
  metrics plan, 31:143
  reusable software requirements and architectures, 28:189
  software engineers' competencies, 28:19
  software salvaging and call dominance tree, 28:117
  specification model, 31:7
  structured analysis and design (SA/SD) methodology vs. text-oriented analysis, 28:49
  U.S. vs. Japan and, 31:109
Software effort estimate alterations, 31:135
Software engineering
  architectural basis for evolving systems, 30:27
  CASE integration, 28:59
  designing testable, heterogenous environments, 29:199
  education, 29:92
  formal methods, 29:95
  leading scholars/institutions, 31:3
  Mythseeker project, 28:89
  quantitative evaluation of process improvement, 28:169
  reuse
    hybrid domain-specific kits, 30:213
    incentive programs, 30:187
    information access tools, 30:231
    measurement of language-supported, 30:271
    modeling reuse across software cycle, 30:295
    REBOOT approach, 30:201
    reusable software libraries, 30:187
    reusable software requirements and architectures, 28:189
    ROSA, 30:249
  software engineers' competencies, 28:19
  software measurement, 29:1
  software salvaging and call dominance tree, 28:117
  systems engineering and management for reengineering, 30:3
  unified metric of software complexity, 29:17
Software engineers' attributes, 28:19
Software maintenance
  composite complexity models and metric for, 31:157
  fault locality and correction, 29:75
  high-risk module detection, 29:85
  industrial software systems and, 29:3
  neural network approach, 29:85, 29:97
  program understanding and transformation tool, 30:99
Software measurement
  academic/industrial collaboration in, 29:1
  evaluation and, 31:171
Software metrics
  composite complexity models and metric for software maintenance, 31:157
  metric-based decision making, 31:95
  metrics plan in software development, 31:143
  for requirements engineering, 29:39
  unified software complexity, 29:17
Software migration, reengineering and, 30:137
Software ownership, 29:191
Software process
  quantitative evaluation of improvement, 28:169
  reengineering, 30:71
Software production
  productivity measurement, 31:71
  static error prediction models, 28:129
Software projects, solution approaches, 31:1
Software quality
  knowledge-based systems and, 29:197
  quality assessment framework for knowledge-based systems, 29:219
Software reengineering
  architectural approaches for software systems, 30:27
  business process, 30:71
  for complex systems, 30:1, 30:45
  data fusion, 30:127
  process, 30:71
  software migration and, 30:137
  software reuse, 28:117, 28:189, 30:117
  systems engineering and management for, 30:3
Software repositories, populating, 30:187
Software reuse
  current status and trends, 30:167
  domain selection/analysis, 30:171
  hybrid domain-specific kits, 30:213
  incentive programs, 30:187
  information access tools, 30:231
  managerial aspects, 30:171
  measurement of language-supported, 30:271
  modeling reuse across software cycle, 30:295
  project experiences, 30:171
  REBOOT approach, 30:201
  reengineering processes, 28:117, 28:189, 30:117, 30:167
  reusable software libraries, 30:187
  ROSA, 30:249
Solution approaches, software projects and different, 31:1
Specification model, concurrent and distributed, 31:7
Static error prediction models, 28:129
Statistical analysis techniques, for software process improvement evaluation, 28:169
Structure charts, information system professionals'attitudes, 28:39
Structured analysis and design (SA/SD) methodology
  CASE tool and, 28:49
  text-oriented analysis vs., 28:49
Structured approaches, research, 28:93
Structured methodology, CASE tools and, 28:49
System monitoring, security and, 31:267
Systems engineering, for reengineering, 30:3
Systems management, for reengineering, 30:3

T

Team capability, software engineers' competencies and, 28:19
Testing strategies, mutation testing, 31:185
Testing tools, for Ada programs, 31:239
Text-oriented analysis, vs. structured analysis and design (SA/SD) methodology, 28:49
Threads of events, designing control systems with entity- life modeling, 28:225
Timed-probabilistic behaviors, model for, 28:239
Timing contraints, validation methods for, 30:85
Transaction-processing architectures, distributed real-time data base systems and, 31:97
Transformation tasks, reengineering complex systems and, 30:45
Transformation tool, program understanding and, 30:99

U

Usability criteria, for automated debugging systems, 31:55

V

Validation methods
for knowledge-based systems, 29:199, 29:235
for real-time constraints, 30:85
Value measurement, unified metric of software complexity, 29:17
Vector-processing elements, distributed memory computers and, 29:107
Verification methods
for knowledge-based systems, 29:199, 29:235
for object-oriented system, 29:261
Visual formalism, for object-oriented conceptual modeling, 31:33
Visual simulation support environment, DOMINO conceptual framework and, 31:215

W

Wrapping approach, for engineering software environments, 29:199